

Ridlo W. Wibowo || 20912009

October 11, 2012

Soal 1. Selesaikan SPL di bawah ini menggunakan metode iteratif:

$$\begin{aligned}
 4x_1 - x_2 \quad \quad - x_4 \quad \quad &= 0 \\
 -x_1 + 4x_2 - x_3 \quad \quad - x_5 &= 5 \\
 \quad \quad -x_2 + 4x_3 \quad \quad - x_6 &= 0 \\
 -x_1 \quad \quad \quad + 4x_4 - x_5 &= 6 \\
 \quad \quad -x_2 \quad \quad - x_4 + 4x_5 - x_6 &= -2 \\
 \quad \quad \quad -x_3 \quad \quad - x_5 + 4x_6 &= 6
 \end{aligned}
 \tag{0.1}$$

SPL di atas dapat diubah dalam bentuk matrix:

$$Ax = b \tag{0.2}$$

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 6 \\ -2 \\ 6 \end{bmatrix}$$

terlihat bahwa matriks A merupakan matriks yang dominan diagonal kuat, sehingga metode iteratif akan selalu mendapatkan solusi untuk semua tebakan awal $x^{(0)}$.

Kita akan menggunakan metode iteratif dalam menyelesaikan SPL ini, yaitu metode Jacobi dan Gauss-Seidel, dengan tebakan awal $x^{(0)} = [0, 0, 0, 0, 0, 0]^T$ dan $\epsilon = 1.0 \times 10^{-5}$

Iterasi Jacobi

Skema iterasi Jacobi:

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k-1)} \right) \quad (0.3)$$

Algoritma iterasi Jacobi:

Input

Vektor solusi tebakan awal $x^{(0)}$, toleransi eps dan iterasi maksimum $Nmax$.

Output

Vektor solusi hingga tercapai toleransi dan jumlah iterasi.

Langkah-langkah

1. Set $xi_i = xo_i$ (tebakan awal) dan $k = 0$
2. Lakukan untuk $xf_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} xi_j \right)$
3. Hitung Infinity Norm dari $(xf_i - xi_i)$ dan (xi_i)
4. Hitung $epsilon = \text{Infinity Norm}(xf_i - xi_i) / \text{Infinity Norm}(xi_i)$
5. $k++$
6. Swap $xi_i = xf_i$
7. Cek apakah $epsilon < toleransi$ dan $k < Nmax$, jika iya, kembali ke langkah 2, jika tidak hentikan program.

Program:

```
#include <iostream>
#include <stdio.h>
#include <math.h>
using namespace std;
int main(){
    int i, j, k=0, Nmax=100000;
    int n=6;
    double A['n']['n']={{4., -1., 0., -1., 0., 0.},
                        {-1., 4., -1., 0., -1., 0.},
                        {0., -1., 4., 0., 0., -1.},
                        {-1., 0., 0., 4., -1., 0.},
                        {0., -1., 0., -1., 4., -1.},
                        {0., 0., -1., 0., -1., 4.}};
    double b['n']={0., 5., 0., 6., -2., 6.};
    double xi['n']={0., 0., 0., 0., 0., 0.};
    double ep=0.00001;
```

```

double xf['n'], eps, sum, vnorm_f['n'], norm_i, norm_f;

cout << "### Program Jacobi untuk menyelesaikan SPL Ax=b ###\n";

cout << "Matriks A : \n";
for (i=0; i<n; i++){
    for (j=0; j<n; j++){
        cout << A[i][j] << "\t";
    }
    cout << "\n";
}
cout << "Vektor b : \n";
for (i=0; i<n; i++){
    cout << b[i] << "\t";
}
cout << endl;
cout << "Tebakan awal x(k=0): \n";
for (i=0; i<n; i++){
    cout << xi[i] << " ";
}
cout << endl;
cout << "epsilon : " << ep << endl;
// Jacobi method
cout << "Nilai barisan solusi x(k) dan epsilon : \n";
do {
    for (i=0; i<n; i++){
        sum = 0.;
        for (j=0; j<n; j++){
            if (j != i){
                sum = sum - A[i][j]*xi[j];
            }
        }
        xf[i] = (b[i] + sum)/A[i][i];
    }

    // menghitung epsilon dari Infinity Norm
    for (i=0; i<n; i++){
        vnorm_f[i] = xf[i] - xi[i];
    }
    norm_f = fabs(vnorm_f[0]);
    norm_i = fabs(xf[0]);
    for (i=1; i<n; i++){
        if (fabs(vnorm_f[i]) > norm_f){norm_f = fabs(vnorm_f[i]);}
        if (fabs(xf[i]) > norm_i){norm_i = fabs(xf[i]);}
    }
    eps = norm_f/norm_i;
    k++;

    for (i=0; i<n; i++){
        xi[i] = xf[i]; // tukar x(k) dengan x(k+1)
        printf("%7.4f ", xf[i]);
    }
    cout << "\t" << eps;
    cout << endl;

} while (eps > ep && k < Nmax);
cout << "Jumlah iterasi : " << k << endl;
return 0;}

```

Hasil:

```
### Program Jacobi untuk menyelesaikan SPL Ax=b ###
Matriks A :
4 -1 0 -1 0 0
-1 4 -1 0 -1 0
0 -1 4 0 0 -1
-1 0 0 4 -1 0
0 -1 0 -1 4 -1
0 0 -1 0 -1 4
Vektor b :
0 5 0 6 -2 6
Tebakan awal x(k=0):
0 0 0 0 0 0
epsilon : 1e-05
Nilai barisan solusi x(k) dan epsilon :
0.0000 1.2500 0.0000 1.5000 -0.5000 1.5000 1
0.6875 1.1250 0.6875 1.3750 0.5625 1.3750 0.772727
0.6250 1.7344 0.6250 1.8125 0.4688 1.8125 0.336207
0.8867 1.6797 0.8867 1.7734 0.8398 1.7734 0.209251
0.8633 1.9033 0.8633 1.9316 0.8066 1.9316 0.115774
0.9587 1.8833 0.9587 1.9175 0.9417 1.9175 0.07041
0.9502 1.9648 0.9502 1.9751 0.9296 1.9751 0.0412546
0.9850 1.9575 0.9850 1.9699 0.9787 1.9699 0.0249648
0.9819 1.9872 0.9819 1.9909 0.9743 1.9909 0.0149087
0.9945 1.9845 0.9945 1.9890 0.9923 1.9890 0.0090067
0.9934 1.9953 0.9934 1.9967 0.9907 1.9967 0.00541521
0.9980 1.9944 0.9980 1.9960 0.9972 1.9960 0.00326949
0.9976 1.9983 0.9976 1.9988 0.9966 1.9988 0.00197056
0.9993 1.9979 0.9993 1.9985 0.9990 1.9985 0.00118949
0.9991 1.9994 0.9991 1.9996 0.9988 1.9996 0.000717555
0.9997 1.9993 0.9997 1.9995 0.9996 1.9995 0.000433102
0.9997 1.9998 0.9997 1.9998 0.9995 1.9998 0.000261352
0.9999 1.9997 0.9999 1.9998 0.9999 1.9998 0.000157742
0.9999 1.9999 0.9999 1.9999 0.9998 1.9999 9.51996e-05
1.0000 1.9999 1.0000 1.9999 1.0000 1.9999 5.74584e-05
1.0000 2.0000 1.0000 2.0000 0.9999 2.0000 3.46784e-05
1.0000 2.0000 1.0000 2.0000 1.0000 2.0000 2.09303e-05
1.0000 2.0000 1.0000 2.0000 1.0000 2.0000 1.26324e-05
1.0000 2.0000 1.0000 2.0000 1.0000 2.0000 7.62435e-06
Jumlah iterasi : 24
```

Iterasi Gauss-Seidel

Skema iterasi Gauss-Seidel:

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k-1)} \right) \quad (0.4)$$

Algoritma iterasi Gauss-Seidel:

Input

Vektor solusi tebakan awal $x^{(0)}$, toleransi eps dan iterasi maksimum $Nmax$.

Output

Vektor solusi hingga tercapai toleransi dan jumlah iterasi.

Langkah-langkah

1. Set $xi_i = xo_i$ (tebakan awal) dan $k = 0$
2. Lakukan untuk $xf_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}xf_j - \sum_{j=i+1}^n a_{ij}xi_j \right)$
3. Hitung Infinity Norm dari $(xf_i - xi_i)$ dan (xi_i)
4. Hitung $epsilon = \text{Infinity Norm}(xf_i - xi_i) / \text{Infinity Norm}(xi_i)$
5. $k++$
6. Swap $xi_i = xf_i$
7. Cek apakah $epsilon < toleransi$ dan $k < Nmax$, jika iya, kembali ke langkah 2, jika tidak hentikan program.

Program:

```
#include <iostream>
#include <stdio.h>
#include <math.h>
using namespace std;
int main(){
    int i, j, k=0, Nmax=100000;
    int n=6;
    double A['n']['n']={{4., -1., 0., -1., 0., 0.},
                        {-1., 4., -1., 0., -1., 0.},
                        {0., -1., 4., 0., 0., -1.},
                        {-1., 0., 0., 4., -1., 0.},
                        {0., -1., 0., -1., 4., -1.},
                        {0., 0., -1., 0., -1., 4.}};
    double b['n']={0., 5., 0., 6., -2., 6.};
    double xi['n']={0., 0., 0., 0., 0., 0.};
    double ep=0.00001;
    double omega=1.1;
    double xf['n'], eps, sum_a, sum_b, vnorm_f['n'], norm_i, norm_f;
    cout << "### Program Gauss-Seidel untuk menyelesaikan SPL Ax=b ###\n";
    cout << "Matriks A : \n";
    for (i=0; i<n; i++){
        for (j=0; j<n; j++){
            cout << A[i][j] << "\t";
```

```

        cout << "\n";}
    cout << "Vektor b : \n";
    for (i=0;i<n;i++){
        cout << b[i] << "\t";}
    cout << endl;
    cout << "Tebakan awal x(k=0): \n";
    for (i=0;i<n;i++){
        cout << xi[i] << " ";}
    cout << endl;
    cout << "epsilon : " << ep << endl;
    // Gauss-Seidel method
    cout << "Nilai barisan solusi x(k) dan epsilon : \n";
    do { // iterasi
        for (i=0;i<n;i++){
            sum_a = 0.;
            sum_b = 0.;
            for (j=0;j<=i-1;j++){
                if (j != i){
                    sum_a = sum_a - A[i][j]*xf[j];}}
            for (j=i+1;j<=n;j++){
                if (j != i){
                    sum_b = sum_b - A[i][j]*xi[j];}}
            xf[i] = (b[i] + sum_a + sum_b)/A[i][i];}

        // menghitung epsilon dari Infinity Norm
        for (i=0;i<n;i++){
            vnorm_f[i] = xf[i] - xi[i];}
        norm_f = fabs(vnorm_f[0]);
        norm_i = fabs(xf[0]);
        for (i=1;i<n;i++){
            if (fabs(vnorm_f[i]) > norm_f){norm_f = fabs(vnorm_f[i]);}
            if (fabs(xf[i]) > norm_i){norm_i = fabs(xf[i]);}
        }
        eps = norm_f/norm_i;
        k++;
        for (i=0;i<n;i++){
            xi[i] = xf[i]; // tukar x(k) dengan x(k+1)
            printf("%7.4f ", xf[i]);}
        cout << "\t" << eps;
        cout << endl;

    } while (eps > ep && k < Nmax); // penghenti iterasi
    cout << "Jumlah iterasi : " << k << endl;
    return 0;
}

```

Hasil:

Program Gauss-Seidel untuk menyelesaikan SPL $Ax=b$ ###
 Matriks A :

```

4 -1 0 -1 0 0
-1 4 -1 0 -1 0
0 -1 4 0 0 -1
-1 0 0 4 -1 0
0 -1 0 -1 4 -1
0 0 -1 0 -1 4
Vektor b :
0 5 0 6 -2 6
Tebakan awal  $x(k=0)$ :
0 0 0 0 0 0
epsilon : 1e-05
Nilai barisan solusi  $x(k)$  dan epsilon :
0.0000 1.2500 0.3125 1.5000 0.1875 1.6250 1
0.6875 1.5469 0.7930 1.7188 0.7227 1.8789 0.365904
0.8164 1.8330 0.9280 1.8848 0.8992 1.9568 0.146226
0.9294 1.9391 0.9740 1.9572 0.9633 1.9843 0.0569653
0.9741 1.9778 0.9905 1.9843 0.9866 1.9943 0.0223799
0.9905 1.9919 0.9966 1.9943 0.9951 1.9979 0.00824212
0.9966 1.9971 0.9987 1.9979 0.9982 1.9992 0.00300691
0.9987 1.9989 0.9995 1.9992 0.9994 1.9997 0.00109549
0.9995 1.9996 0.9998 1.9997 0.9998 1.9999 0.000399051
0.9998 1.9999 0.9999 1.9999 0.9999 2.0000 0.000145362
0.9999 1.9999 1.0000 2.0000 1.0000 2.0000 5.29515e-05
1.0000 2.0000 1.0000 2.0000 1.0000 2.0000 1.92889e-05
1.0000 2.0000 1.0000 2.0000 1.0000 2.0000 7.02649e-06
Jumlah iterasi : 13

```

Kita dapat melakukan sedikit perubahan pada metode Gauss-Seidel ini agar lebih cepat konvergen. Banyak kasus yang terjadi pada metode iteratif yakni perubahan x_i selalu menuju ke arah yang sama (tidak berosilasi). Sehingga *over-correcting* (*over-relaxing*) untuk x_i dengan nilai yang tepat dapat mempercepat kekonvergenan. Hal ini dapat diilustrasikan seperti gambar di bawah ini.

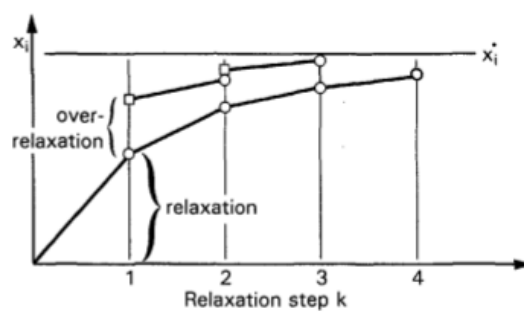


Figure 0.1: Over-relaxation.

Dengan memasukkan faktor over-relaksasi ω (biasanya $1.0 < \omega < 2.0$) maka kita peroleh metode yang kemudian biasa disebut sebagai metode *Successive Over-Relaxation* (SOR).

Skema SOR :

$$x_i^{(k)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right) \quad (0.5)$$

dengan menggunakan skema di atas, kita dapat melakukan sedikit perubahan pada program Gauss-Seidel di atas. Untuk kasus yang sama dengan $\omega = 1.1$ diperoleh hasil iterasi:

```
### Program Gauss-Seidel + Successive Over-Relaxation ###
Tebakan awal x(k=0):
0 0 0 0 0 0
epsilon    : 1e-05
omega SOR : 1.1
Nilai barisan solusi x(k) dan epsilon :
0.0000  1.3750  0.3781  1.6500  0.2819  1.8315  1
0.8319  1.6478  0.9190  1.7913  0.8712  1.9592  0.424609
0.8626  1.9397  0.9803  1.9477  0.9707  1.9906  0.146671
0.9828  1.9878  0.9960  1.9924  0.9949  1.9985  0.0601595
0.9963  1.9977  0.9993  1.9983  0.9990  1.9997  0.00675739
0.9993  1.9996  0.9999  1.9997  0.9998  1.9999  0.00149757
0.9999  1.9999  1.0000  1.9999  1.0000  2.0000  0.000293119
1.0000  2.0000  1.0000  2.0000  1.0000  2.0000  5.2646e-05
1.0000  2.0000  1.0000  2.0000  1.0000  2.0000  9.94652e-06
Jumlah iterasi : 9
```

Analisis hasil dan kesimpulan

Solusi eksak untuk SPL di atas adalah $x_i = [1, 2, 1, 2, 1, 2]^T$ dan dari program di atas terlihat bahwa:

1. Metode Gauss-Seidel lebih cepat konvergen dibandingkan dengan metode Jacobi. Metode Jacobi membutuhkan 24 kali iterasi sedangkan Gauss-Seidel hanya membutuhkan 13 kali iterasi, dengan toleransi yang sama yaitu 1.0×10^{-5} .
2. Sedikit penambahan pada metode Gauss-Seidel kita peroleh skema *Successive Over-Relaxation* (SOR), yang apabila kita memasukan ω yang tepat maka akan mempercepat kekonvergenan metode iterasi (9 kali iterasi dengan nilai $\omega = 1.1$ untuk nilai toleransi yang sama).

Soal 2. Terdapat Sistem Persamaan Linier (SPL) sebagai berikut:

$$\begin{aligned} 2x_1 - x_2 + x_3 &= -1 \\ 2x_1 + 2x_2 + 2x_3 &= 4 \\ -x_1 - x_2 + 2x_3 &= -5 \end{aligned} \tag{0.6}$$

yang mempunyai solusi eksak $x_i = [1, 2, -1]^T$

1. Tunjukkan bahwa $\rho(Tj) = \frac{\sqrt{5}}{2} > 1$.
2. Tunjukkan bahwa metode iterasi Jacobi dengan hampiran awal $x_i^{(0)} = 0$ akan gagal menghasilkan hampiran yang baik setelah 25 iterasi.
3. Tunjukkan bahwa $\rho(Tg) = \frac{1}{2}$.
4. Pergunakan iterasi Gauss-Seidel dengan hampiran awal $x_i^{(0)} = 0$ untuk mengjampiri solusi SPL dengan toleransi 10^{-5} dalam norm tak hingga (L_∞).

Jawab:

SPL dalam bentuk matriks $Ax = b$:

$$\begin{bmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 4 \\ -5 \end{bmatrix}$$

1. Tunjukkan $\rho(Tj) = \frac{\sqrt{5}}{2} > 1$

$Tj = D^{-1}(L + U)$, dengan

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$

sehingga diperoleh

$$\begin{aligned} D^{-1} &= \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \\ Tj = D^{-1}(L + U) &= \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix} \right) \end{aligned}$$

$$Tj = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

$$\det(Tj - \lambda I) = \begin{vmatrix} -\lambda & \frac{1}{2} & -\frac{1}{2} \\ -1 & -\lambda & -1 \\ \frac{1}{2} & \frac{1}{2} & -\lambda \end{vmatrix} = -\lambda^3 - \frac{5}{4}\lambda = -\lambda(\lambda^2 + \frac{5}{4})$$

di peroleh: $\lambda_1 = 0$, $\lambda_2 = -\sqrt{\frac{5}{4}}$, $\lambda_3 = \sqrt{\frac{5}{4}}$

maka $\rho(Tj) = \max\{|0|, |-\sqrt{\frac{5}{4}}|, |\sqrt{\frac{5}{4}}|\} = \sqrt{\frac{5}{4}} > 1$ (terbukti).

2. Dengan menggunakan program pada soal sebelumnya untuk kasus di atas, diperoleh output hingga iterasi ke 25 seperti di bawah ini (tidak konvergen walaupun jumlah iterasi ditambah).

```
### Program Jacobi untuk menyelesaikan SPL Ax=b ###
Matriks A :
2 -1 1
2 2 2
-1 -1 2
Vektor b :
-1 4 -5
Tebakan awal x(k=0):
0 0 0
epsilon : 1e-05
Nilai barisan solusi x(k) dan epsilon :
-0.5000 2.0000 -2.5000 1
1.7500 5.0000 -1.7500 0.6
2.8750 2.0000 0.8750 1.04348
0.0625 -1.7500 -0.0625 2.14286
-1.3438 2.0000 -3.3438 1.1215
2.1719 6.6875 -2.1719 0.700935
3.9297 2.0000 1.9297 1.19284
-0.4648 -3.8594 0.4648 1.51822
-2.6621 2.0000 -4.6621 1.25681
2.8311 9.3242 -2.8311 0.785505
5.5776 2.0000 3.5776 1.31314
-1.2888 -7.1553 1.2888 1.27951
-4.7220 2.0000 -6.7220 1.36198
3.8610 13.4441 -3.8610 0.851236
8.1526 2.0000 6.1526 1.40374
-2.5763 -12.3051 2.5763 1.16253
-7.9407 2.0000 -9.9407 1.43905
5.4703 19.8814 -5.4703 0.899403
12.1759 2.0000 10.1759 1.46859
-4.5879 -20.3517 4.5879 1.09827
-12.9698 2.0000 -14.9698 1.49312
7.9849 29.9397 -7.9849 0.933199
```

18.4623 2.0000 16.4623 1.51334
 -7.7311 -32.9246 7.7311 1.06074
 -20.8279 2.0000 -22.8279 1.52991
 Jumlah iterasi : 25

3. Tunjukkan $\rho(Tg) = \frac{1}{2} < 1$ (akan konvergen)

$Tg = (D - L)^{-1}U$, dengan

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$

sehingga diperoleh

$$D - L = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & -1 & 2 \end{bmatrix}, (D - L)^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

$$Tg = (D - L)^{-1}U = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Tg = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

$$\det(Tg - \lambda I) = \begin{vmatrix} -\lambda & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} - \lambda & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} - \lambda \end{vmatrix} = -\lambda \left(\frac{1}{2} + \lambda\right)^2$$

di peroleh: $\lambda_1 = 0$, $\lambda_2 = -\frac{1}{2}$, $\lambda_3 = \frac{1}{2}$

maka $\rho(Tg) = \max\{|0|, |-\frac{1}{2}|, |\frac{1}{2}|\} = \frac{1}{2} < 1$ (terbukti).

4. Dengan menggunakan program Gauss-Seidel pada soal sebelumnya untuk nilai toleransi 10^{-5} menggunakan l_{∞} diperoleh hasil seperti di bawah ini.

```
### Program Gauss-Seidel untuk menyelesaikan SPL Ax=b ###
Matriks A :
2 -1 1
2 2 2
-1 -1 2
Vektor b :
-1 4 -5
Tebakan awal x(k=0):
0 0 0
epsilon : 1e-05
Nilai barisan solusi x(k) dan epsilon :
-0.5000 2.5000 -1.5000 1
```

1.5000	2.0000	-0.7500	1
0.8750	1.8750	-1.1250	0.333333
1.0000	2.1250	-0.9375	0.117647
1.0312	1.9062	-1.0312	0.114754
0.9688	2.0625	-0.9844	0.0757576
1.0234	1.9609	-1.0078	0.0517928
0.9844	2.0234	-0.9961	0.030888
1.0098	1.9863	-1.0020	0.0186824
0.9941	2.0078	-0.9990	0.0107004
1.0034	1.9956	-1.0005	0.00611696
0.9980	2.0024	-0.9998	0.0034138
1.0011	1.9987	-1.0001	0.00189336
0.9994	2.0007	-0.9999	0.00103722
1.0003	1.9996	-1.0000	0.000564687
0.9998	2.0002	-1.0000	0.000305143
1.0001	1.9999	-1.0000	0.000164041
0.9999	2.0001	-1.0000	8.77354e-05
1.0000	2.0000	-1.0000	4.67308e-05
1.0000	2.0000	-1.0000	2.47953e-05
1.0000	2.0000	-1.0000	1.31131e-05
1.0000	2.0000	-1.0000	6.91412e-06

Jumlah iterasi : 22

Hasil di atas menunjukkan bahwa dengan menggunakan iterasi gauss-seidel solusinya konvergen menuju solusi eksak untuk toleransi 10^{-5} setelah iterasi ke 22.

Soal 3. Carilah akar-akar dari sistem persamaan nonlinier berikut ini menggunakan metode Newton.

$$f_1(x_1, x_2) = 0.5 \sin(x_1 x_2) - 0.25 \frac{x_2}{\pi} - 0.5x_1 \quad (0.7)$$

$$f_2(x_1, x_2) = \left(1 - \frac{0.25}{\pi}\right)(\exp(2x_1 - e) + e \frac{x_2}{\pi} - 2ex_1) \quad (0.8)$$

Jika kita gambar 2 fungsi di atas untuk $z = 0$ dengan $-1 < x < 3$ dan $-20 < y < 5$ maka diperoleh kurva seperti berikut:

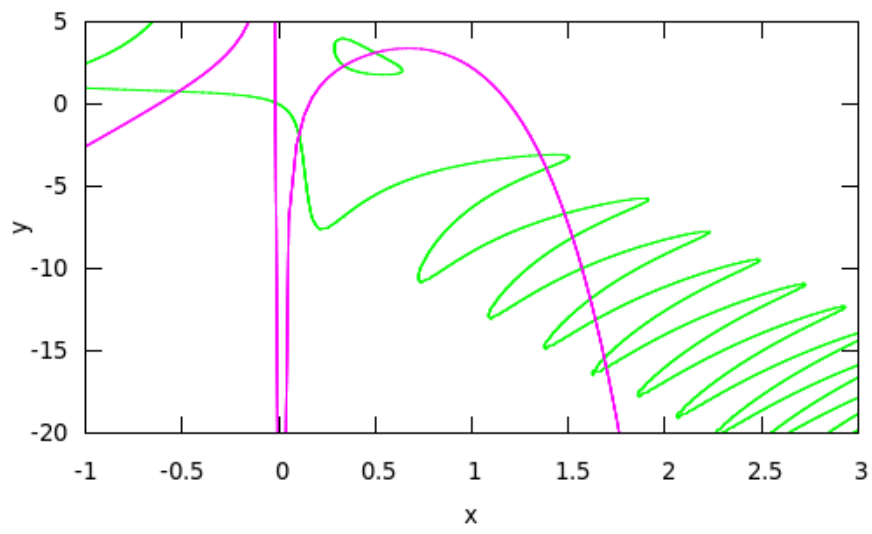


Figure 0.2: $f_1(x_1, x_2)$ dan $f_2(x_1, x_2)$.